

AMIGA

\$2
RRP

WORKBENCH

Registered by Australia Post - Publication No. VBG 7930

Number 34

Circulation:1300

March 1989



Next AUG Meeting
Sunday, March 19th, 1989 at 2pm

(Doors open at 1pm, meeting starts at 2pm sharp)

**AUG meetings are held at Victoria College Burwood Campus
Burwood Highway, Burwood Melways map 61 reference B5.**

Amiga Users Group Inc, PO Box 48, Boronia, 3155, Victoria, Australia

Australia's Largest Independent Association of Amiga Owners
The *Amiga Users Group Inc* has no affiliation with Commodore
Amiga is a trademark of Commodore-Amiga, Inc

AMIGA™ Users Group

Post Office Box 48, Boronia, 3155, Victoria

Who Are WE?

The Amiga Users Group is a not-for-profit association of people interested in the Amiga computer and related topics. With over 1000 members, we are the largest independent association of Amiga users in Australia.

Club Meetings

Club meetings are held at 2pm on the third Sunday of each month at Victoria College, Burwood Highway, Burwood. Details on how to get there are on the back cover of this newsletter. The dates of upcoming meetings are:

Sunday, March 19th at 2pm

Sunday, April 16th at 2pm

Sunday, May 21st at 2pm

Production Credits

This month's newsletter was edited by Con Kolivas. Equipment and software used was: Amiga 500, Professional Page, Excellence!, and Apple LaserWriter Plus.

Copyright and Reprint Privileges

Amiga Workbench is copyright 1989 by the Amiga Users Group Inc. Articles herein that are copyrighted by individual authors or otherwise explicitly marked as having restricted reproduction rights may not be reprinted or copied without written permission from the Amiga Users Group or the authors. All other articles may be reprinted for any non-commercial purpose if accompanied by a credit line including the original author's name and the words "Reprinted from Amiga Workbench, newsletter of the Amiga Users Group, PO box 48, Boronia, 3155".

Contributions

Articles, papers, drawings, cartoons and comments are actively sought for publication in Amiga Workbench. All contributions submitted for the purpose of publication that are printed in the newsletter are rewarded on the basis of one free public domain disk copy per column or half page printed with a minimum of one free copy. Contributions may be sent in on disk, paper or uploaded to Amiga Link or Amiga Link II in the area set aside for this purpose. Please send your contributions in text only, non-formatted if they are on file and remember to include your address for return of disks and tokens for PD disks. Absolute deadline for articles is 16 days before the meeting date. Contributions can be sent to: The Editor, AUG, PO box 48, Boronia, 3155.

Membership and Subscriptions

Membership of the Amiga Users Group is available for an annual fee of \$25. To become a member of AUG, fill in the membership form in this issue (or a photocopy of it), and send it with a cheque or money order for \$25 to: Amiga Users Group, PO Box 48, Boronia, 3155

Public Domain Software

Disk from our public domain library are available on quality 3.5" disks for \$8 each including postage on AUG supplied disks, or \$2 each on your own disks. The group currently holds over 200 volumes, mostly sourced from the USA, with more on the way each month. Details of latest releases are printed in this newsletter, and a catalog disk is also available.

Member's Discounts

The Amiga Users Group negotiates discounts for its members on hardware, software and books.

Currently, Technical Books in Swanston Street in the city offers AUG members a 10% discount on computer related books, as does McGills in Elizabeth Street. Ultraphase are also willing to give substantial discounts. Just show your membership card. Although we have no formal arrangements with other companies yet, most seem willing to offer a discount to AUG members. It always pays to ask!

Back Issues of Workbench

All back issues of Amiga Workbench are now available, for \$2 each including postage. Note that there may be delays while issues are reprinted. Back issues are also available at meetings.

Amiga Link I & II - Our Bulletin Board Systems

The Amiga Users Group operates two bulletin board systems devoted to the Amiga, using the Opus message and conferencing software. AmigaLink I and II are available 24 hours a day. Both AmigaLink I & 2 can be accessed at V21 (300bps), V22 (1200bps), V23 (1200/75bps) and V22bis (2400bps) using 8 data bits, 1 stop bit and no parity. Phone numbers for Amigalink are:

Amigalink I - (03) 792 3918

Amigalink II - (03) 376 6385

AmigaLink is part of a world-wide network of bulletin boards, and we participate in national and international Amiga conferences. AmigaLink has selected Public Domain software available for downloading, and encourages the uploading of useful public domain programs from its users. AmigaLink I is OzNet node number 8:830/324 and AmigaLink II is OzNet node number 1305/998

Newsletter Advertising

The Amiga Users Group accepts commercial advertising in Amiga Workbench subject to the availability of space at these rates:

Quarter page	\$20
Half page	\$40
Full page	\$70

These rates are for full-size camera-ready copy only. We have no photographic or typesetting facilities. Absolute deadline for copy is 23 days before the meeting date. Send the copy and your cheque to: The Editor, AUG, PO Box 48, Boronia, 3155, Victoria.

Amiga Users Group Committee

Co-ordinator:	Position Vacant		
Ass. Co-Ord:	Lester McClure	233 5664	Mt. Waverley
Meeting Chair:	Ron Wail	848 7283	Doncaster
Secretary:	John Elston	375 4142	Moonee Ponds
Treasurer:	Donna Heenan	729 2327	Bayswater
Membership:	Neil Murray	794 5683	Dandenong
Purchasing:	Bohdan Ferens	792 1138	Dandenong
Book Library:	Ross Johnson	824 7026	Malvern
Disk Library:	Craig Hutchison	484 1339	Carnegie
Editor:	Con Kolivas	428 0682	Thornbury
Committee:	Doug Myers	752 2053	Abbottsford
	Peter Jetson		

Amiga Workbench

Number 34

March 1989

Page 2

MY HOLIDAYS

by Bruce Warren

I've been meaning to write an article for WORKBENCH for some time now, in fact I did write one in May 1988 and never sent it in, for reasons explained below. It was partly the appearance of articles in the December 1988 and January 1989 WORKBENCH'S and partly because I had a few weeks holiday to play around on my AMIGA that provided the incentive and ammunition to write the following short articles.

1. VIRUSES

In Darren King's article on Viruses in the January 1989 WORKBENCH he raised the question about viruses which modified the time in the real-time clock. Actually I just stopped writing to reread the paragraph and realised that I had misinterpreted it the first time, to mean viruses which reside in the battery backed RAM of the real-time clock. I think the comment I was going to make is still worth making though, and that is that such a virus cannot exist. My source of information is a book in a series on the Amiga, published by ABACUS. I can't recall the exact title but it was something like; "AMIGA DISK DRIVES - INS AND OUTS" and it is book 9 in the series. The comment made was to the effect that there is only about 1K of battery backed RAM in the real-time clock, which apparently is not enough. The other point made was that even if it was enough memory, when you turn the power off, any vector initialized to point to this memory is lost. Just getting back to Darren's original question brings me to my other comments on VIRUSES. Only a few days after reading the article a friend of my wife's rang to say they had just bought some AMIGA software and after making backup copies of originals, two programs did not work. Firstly she was new to computing and was unaware of copy protection. The documentation which came with the AMIGA had proposed making backups of all disks using "Diskcopy". For any newcomers reading this article, not all programs can be copied using "Diskcopy" (Although some may appear to copy and when you try to use it you are asked to insert the original). This was not the problem in this case. The problem was, as you cleverly guessed, a VIRUS.

A quick check through all her disks revealed a BYTE-BANDIT VIRUS on about 90 percent of her disks. I cleaned up most of the disks, but 2 had to be returned to the retailer since the VIRUS had

ruined the bootblock. All software had been purchased from just 2 shops and none were public domain software. In other words the retail stores are also passing the VIRUS.

Anyway, after passing on a few precautionary tactics (ie. leave write protect tags in the "protect" position.) I noticed that the write protect tag on my own Workbench copy was off. Just for laughs I did a very quick check for a VIRUS. The quick check, in case you are wondering actually indicates that a VIRUS may be in memory. Therefore the procedure is:

i. Start to boot with the suspect disk (only need wait a few seconds to make sure the bootblock has been read).

ii. Reboot using the CNTRL + Left AMIGA + Right AMIGA keys while holding the left mouse key down at the same time.

iii. Release the keyboard keys while continuing to hold the left mouse button down.

If the screen turns green, after a few seconds, chances are you have a VIRUS. [Ed's note - this only works on variations of the SCA virus!]

Well to cut a long story short I had a VIRUS on 3 disks but it was not the BYTE-BANDIT VIRUS. In fact it wasn't any of those discussed in the previously mentioned article. As I don't have a modem I was unable to pass on the details. The VIRUS seems to go by the name "2Z" and is spread by Micro-Master of CCW & Odie from AEK. I'll try to get access to a modem to pass on some details.

Finally, getting back to the real-time clock, I don't think the above VIRUS resets the clock. However, when I realised what Darren's comment meant I suddenly remembered noticing a few days back that the displayed date was incorrect. Instead of 1989 it was showing 1933. I immediately thought I was hot on the trail of a big time discovery. I reset the date, rebooted with a VIRUS infected disk, checked the date ANDnothing was wrong. The test is inconclusive of course.

[Ed's note - there are programs that can make your clock run at an accelerated rate or reset your clock or disable it completely! A public domain program - chronos can fix most software dysfunctions of your clock]

2. THE EXECUTE() FUNCTION

I refer to Mark Kelly's SIDETRACK on page 6 of the December 1988 issue of WORKBENCH. I sus-

pect the editor was inundated with answers to the problem. The code in the article is correct (except of course the missing brackets after the word "main"). I'm almost positive the problem is that you do not have the "RUN" command in the c directory. Probably of equal interest is my source of information. The EXECUTE() function is a DOS function, therefore it is described, along with the above requirement, in the AMIGA DOS MANUAL Page 189, Bantam Books.

3. AMIGA to IBM

I read Mark Kelly's article on page 6 of the December 1989 issue of WORKBENCH with considerable interest (ie "Amiga to Amstrad & back again"). I've had an IBM for several years and upgraded to AMIGA just over 1 year ago. I've had intentions of cabling up a Null Modem for quite a while for three reasons; firstly, to pass info between AMIGA and IBM, (for example I don't have a modem but I have access to IBM's with modems) secondly, to try out a few of the multi-user games between two AMIGA'S (eg. FLT. SIM.) and thirdly, to save moving the printer back and forth all the time between the IBM and AMIGA (different rooms). I have a parallel interface on the printer but I have a few ideas regarding the use of print files.

When I saw the article I thought the time was ripe. Having bought the connectors months ago I went in search of cable. DICK SMITH ELECTRONICS had 2 types of cable which would do the job. However JAYCAR had 6 core shielded cable at \$1.65 a metre, which was cheaper than DICK SMITH's 4 core. If you intend going to TANDY you probably have a MACINTOSH and don't care about price. I think the connectors cost about \$3 or \$4 each including backshells, which are usually sold separately. Before wiring up the cable I thought I'd check the connections proposed in the above mentioned article against some info in one of my books. Something seemed wrong. I thought best to stick with the book. It did not work. So I tried the other, which also did not work. Time to use my brain I thought.

My approach was to look at what the function of each connection was and then simulate the required conditions at the appropriate interface. The problem of course was with the handshake lines. I think its worth mentioning the funtions here briefly. The serial interface standard used is called RS-232C (Note: RS-232C and CCITT V.24 are equivalent). I should point out that these are my

interpretations from reading various conflicting documents and may not be totally accurate. Anybody knowing better please write an article. RS-232C represents 1's (or ON) by +3 to +20 volts, and 0's (or OFF) by -3 to -20 volts. Here are the main handshake lines:

DTR - Data terminal ready (turned ON by terminal when terminal ready).
 DSR - Data set ready (turned ON by modem when modem ready).
 RTS - Ready to send (turned ON by terminal when it wants to send data).
 CTS - Clear to send (turned ON by modem in response to RTS).
 CD - Carrier detect (turned ON by modem when it detects a carrier signal on the transmission line).

Firstly, consider a typical situation where at location A, terminal-A is connected to modem-A by an RS-232C interface, which in turn is connected via a transmission line (direct with no telephone exchange network) to modem-B (at location B) which is connected to terminal-B by an RS-232C interface. As you can see DTR & DSR are a complimentary pair for each device to tell the other co-located device it is powered up and ready. Similarly RTS & CTS are a complimentary pair, although unlike the previous pair, this pair would only be required at the send end and CTS is a response to RTS, whereas there is no sequence for the DTR/DSR pair. The RTS/CTS pair are intended to facilitate Half-Duplex operation and in our case can be permanently left ON for Full-Duplex operation.

V.24 states that DTR,DSR,RTS and CTS must all be ON before the terminal sends data to the modem. So here, as I see it, is a typical (cutdown) handshake sequence:

- i. All terminals and modems at both location A and B power up. DTR and DSR may be set ON at this point. I found that for both the AMIGA and IBM, DTR is set ON when the communication software is loaded and initializes the port.
- ii. Terminal-A wants to send data, so it sets RTS ON.
- iii. Modem-A, upon detecting RTS ON, generates a carrier signal on the transmission line. After a time delay Modem-A sets CTS ON.
- iv. Modem-B, upon detecting the carrier signal, sets CD ON.

The connection is now ready to carry data from A to B. For a Full-Duplex operation the above could take place in both directions simultaneously. To simulate the line conditions, established above, for the situation where you wish to eliminate the modems (ie. NULL Modem), the following connections provide one solution:

Terminal-A Plug		Terminal-B Plug
TXD	2	3 RXD
RXD	3	2 TXD
DTR	28	6 DSR
DSR	6	28 DTR
RTS	4	8 CTS
CTS	5	4 RTS
CD	8	5 CTS
GND	7	7 GND

The curly braces represent pin 4 connected to pin 5. As you can see, there are 7 cable connections shown. The GND is connected to the cable shielding. It depends on what software you use as to whether the connection to CD is needed. If you don't have the spare wires you could connect pin 8 to pin 20 within the same plug.

Just one small point about plug connectors. If you have an AMIGA 1000 you need a male 25 pin plug. For AMIGA 500 & 2000 you need a female 25 pin plug. For IBM-XT you probably need a 25 pin female. For an IBM-AT you probably need a 9 pin female. Don't you love standards.

The 9 pin connector uses the following pin assignments:

CD pin 1, RTS pin 4, GND pin 5, DSR pin 6, RTS pin 7 and CTS pin 8.

I did intend saying something about software, but I've rambled on a lot longer than initially intended. A quick word perhaps. If you try using the BASIC program proposed by Mark in the December article you may have problems on the IBM as I did. My IBM BASIC is an older version than AMIGA BASIC. The most obvious difference is my IBM BASIC needs line numbers. Most problems show up when you try to run the program and you get interpreter errors. One such line was:

"OPEN filespec\$ AS #serial LEN=buffer"

It was the "LEN" which initially gave problems, so I loaded BASIC using the /C:combuffer specification. Having got the program running I had a lot of trouble sending data in the AMIGA to IBM direction. Usually after sending about 7K of data the IBM would start displaying graphic symbols that look similar to the "!" character between text eventually resulting in both computers freezing up or the IBM timing out. The simplest solution was to use software which uses the XMODEM protocol. The other thing worth mentioning is that I always had to run the BASIC program on the AMIGA before I ran the one on the IBM. Other-

wise the IBM would timeout while initializing the port.

4. AMIGA 500 PROBLEMS

As mentioned at the begining, I wrote an article in MAY 1988 which I did not submit. The following is a modified version of that article:

PLAYING PATIENTS.

By B. Warren.

After holding back for several months, I finally took the plunge around mid december 1987. I bought an AMIGA 500 with 512K RAM expansion unit and the Commodore 1084 monitor. As I type this article it is now the end of May and those mathematicians among you are probably saying to yourselves 5 and a half months of bliss. WRONG. OK, you say, perhaps you had a few problems then, lets try 5 months. WRONG. How about 4 months. WRONG. Well surely 3 months. WRONG, WRONG, WRONG. Is it Rumplestiltskin? Getting warm. To tell the truth I'm not exactly sure, but it would be something like 2 to 2 & a half months. In all it has been in for repair four times. The first time in Jan/Feb for 2 weeks. The second, third and fourth times totalled a further 12 weeks over March/April/May. (It was allowed home on 2 weekends for good behavior). In all 14 weeks I've been without my AMIGA.

So what have the problems been? Well to cut a long story short I'll just list each problem briefly as follows :

Disk drive (failed to validate disks) -- 3 reportings.
 Monitor (flashes on screen) -- 2 reportings.
 Mouse (buttons release prematurely) -- 2 reportings.
 Keyboard (typed incorrect character) -- 1 reporting.

With the exception of the mouse problem the other faults were all intermittent and on the third visit to the doc the disk drive fault refused to show itself. Until, of course, 5 minutes after I got it home. As it turned out it was finicky about some disks only. Fortunately, one of the disks which exhibited the problem (sometimes), was the Basic Extras disk which came with the computer. So I knew it wasn't just the cheap disks I had been buying. To prove I wasn't just dreaming ,and that I didn't get some kind of thrill from lending my AMIGA to some stranger for months on end, I provided 5 of my own disks which had previously caused problems. I've just today picked up my AMIGA for the fourth, and hopefully last time. Now, when I arri-

ved home and opened the paper bag with my disks in. Guess how many disks were inside? For those of you who guessed less than 5 take a bow. For those of you who guessed 5 had better reread this article as self punishment.

Although, in the midst of all this stuffing around Commodore did replace a considerable percentage of my unit (mouse twice, monitor, power supply and apparently a considerable proportion of the computer including disk drive). The most annoying aspect of this whole affair has been the petty penny pinching attitude of Commodore. By this I'm referring to their attitude to supply of spare parts to Victoria. Three of the four times my computer was in for repair, replacement parts were needed. On each occasion they had to be sent from Sydney. And what makes it even worse is, if the unit is still under warranty, they will not send the replacement part until the faulty item has been received. Because they normally use Australia Post, we're talking weeks.

OK, that was back in MAY. The reason I did not submit the above article was; as I finished writing it the monitor did its trick. Well as it turned out I just couldn't be bothered taking it back again, since I had already been without an AMIGA for so long. The monitor still acts up. The problem is an audible click often accompanied by a quick flick on the screen. Usually the flick on the screen is a bright (white) spot, although lately its occassionaly showing up as dark lines across the full width of the screen (sometimes 2 characters in width), usually accompanied by a minor jump in the screen image. I borrowed my brother-in-law's AMIGA 500 with monitor some months ago and experienced the same white flick. I concluded it must be my environment. (Although my brother-in-law has experienced it less frequently). I built a mains power line filter which had no effect. Anybody got any helpful suggestions? FLICK. The mouse has also played up again recently, for a few weeks, before coming good.

5.FAERY TALES

-Hint for exploring mazes - No need for maps, just keep left all the time. Or, alternatively keep right all the time. This way you will cover all possibilities.

-Hint about ring of stones. (don't read if you want to find out for your self). Stand in circle, use blue stone, after screen flashes check the scenery around ring.

-How do the BIRD TOTEMS work?

NWAUG NWAUG NWAUG NWAUG NWAUG
 W U
 A A North West Amiga Users Group
 U W
 G N
 U W
 A A Meetings Held:
 W U
 N G
 W U
 A A Commencing 7:30 pm
 U W
 G N
 U W
 A A Rooms 19 & 20, 1st Floor
 W U
 N G
 W U
 A A Essendon Community Centre
 W U
 N G
 W U
 A A Cnr. Mt Alexander & Pascoe Vale rds.
 W U
 A A Moonee Ponds 3039
 U W
 G N
 U W
 A A Meetings Scheduled
 W U
 A A 29/3/89 5/4/89 12/4/89
 W U
 NWAUG NWAUG NWAUG NWAUG NWAUG

THE AMIGA, THE COPPER, SCROLLING TEXT, AND OTHER NEAT THINGS

by Simon Bullen

Scrolling text smoothly is the object of many an Amiga programmer's dream, it is also an IBM programmer's nightmare. I recently saw a friend's attempt at scrolling on his IBM AT clone (it was pathetically jerky), "And how much processor power are you using for that?" I sniggered. "How do those Amiga game loaders do it?" you wonder.

Many use the blitter to scroll the text across, the programs for these almost always need to be written in 68000 (if they wish to be able to do much else at the same time), and incredibly well written (if you count doing it the hard way well written). The people who do scrolling with the blitter obviously do not know as much about the Amiga as they should - the Amiga has an incredibly powerful chip called The Copper (short for co-processor) which can do scrolling with a minimum of effort. This chip has not really been publicised by Commodore, and is covered in no detail whatsoever in most technical manuals. (The only information I have is from a 1986 hardware manual). The copper can change any of the system registers when the video beam is at almost any position on the screen (horizontal resolution is not very high, but vertical is the same as the pixel

display). In this article I will attempt to explain use of the copper in an "exclusive" situation, where the program will completely take over the monitor, ignoring everybody else's screens and windows - just displaying what we want.

A NOTE ON THIS ARTICLE

The information in this article is on the copper, and is presented in such a way that is useful to you no matter what language you are using, and the C macros are really quite simple to translate into any language you care to.

A SHORT PARAGRAPH OR TWO ON BASIC COPPER PRINCIPLES

To use the copper, you must first write a program for the copper to execute when the video beam scans the display. This program for the copper is called The Copper List. The Copper recognises three different commands in it's list, (3! wow!), these are MOVE, WAIT, and SKIP. (I will probably not cover SKIP in this article - it is not as straightforward as everything else, and has only a limited application).

The MOVE command tells the copper to MOVE some data (of length 1 word) into one of the system registers (the system registers are listed in cop.h, the hardware manual, and an issue of Amiga World)

The WAIT command tells the copper to wait until the video beam gets to the specified location. (remember the video beam is the paintbrush that scans the display every frame drawing up what you see on your monitor. It starts at the top left corner, moves across, and when it hits the edge it drops down one line and back to the left side and paints towards the right. [Much like typing on a piece of paper])

Using MOVE and WAIT together allows us to "wait" for any part of the display, and then change something important - like the palette.

32 of the system registers are the color registers, or the palette - here is an example of how the copper can change the palette anywhere on the screen:

```
COPHIT(c1, 0,100); // wait for line 100 //
COPINVB(c1,COLOR00,0xffff); // change the background color to white //
COPHIT(c1, 0,110); // wait for line 110 //
COPINVB(c1,COLOR00,0x0000); // change the background color the black //
```

A FEW WORDS ABOUT INTUITION USER

COPPER LISTS

When you use the system copper list calls (CMOVE, CINIT...) intuition builds a rather large linked list of rather large structures, and then before you can do anything with them they need to be processed and sorted and turned into a REAL copper list. There is a reason for intuition doing this - it means that your copper instructions can be easily (if not speedily) MERGED into the system copper list - so that you can make additions to system screens. This is fine if you just wanna chuck some color bars across the screen and not move 'em about too much, cos sorting this huge pile of ---- into something meaningful takes a hell of a lot of time (well not really, but when you need to do it every frame (read 50 times a second), like in a scroll...)

MORE THAN A FEW WORDS ABOUT REAL COPPER LISTS

What needs to be done is to build a real copper list in the first place. A real copper instruction comprises of 2 words, an intuition copper instruction structure has about ten variables, which include nested structures - when you have several hundred copper instructions to handle every frame... it slows life down a lot if you play with massive structures instead of little words.

Here is the diagram of a REAL copper instruction:

BIT#	Move		Wait		Skip	
	WORD1	WORD2	WORD1	WORD2	WORD1	WORD2
15	X	RD15	VP7	BFD	VP7	BFD
14	X	RD14	VP6	VE6	VP6	VE6
13	X	RD13	VP5	VE5	VP5	VE5
12	X	RD12	VP4	VE4	VP4	VE4
11	X	RD11	VP3	VE3	VP3	VE3
10	X	RD10	VP2	VE2	VP2	VE2
9	X	RD9	VP1	VE1	VP1	VE1
8	DR8	RD8	VP0	VE0	VP0	VE0
7	DR7	RD7	HP8	HE8	HP8	HE8
6	DR6	RD6	HP7	HE7	HP7	HE7
5	DR5	RD5	HP6	HE6	HP6	HE6
4	DR4	RD4	HP5	HE5	HP5	HE5
3	DR3	RD3	HP4	HE4	HP4	HE4
2	DR2	RD2	HP3	HE3	HP3	HE3
1	DR1	RD1	HP2	HE2	HP2	HE2
0	0	RD0	1	0	1	1

KEY

X = don't care, but should be a 0 for upward comp.
 DRx = destination address (eg BPLCOHL)
 RDxx = RAM data to be moved to destination register
 VPx = Vertical beam position bit
 HPx = Horizontal beam position bit
 VEx = enable comparison (this is a mask for the vertical position... don't worry about it... I don't.)
 HEx = see VE
 BFD = blitter finished disable..

(this table was taken from page 25 of the chapter entitled Coprocessor Hardware, from some manual whose name I can't remember, and which you don't need to know because I have reproduced the whole table)

The table describes the format of the copper instructions that the copper ACTUALLY executes. So to build a directly executable copper list, all we have to do is get a chunk of CHIP memory, and build a list using the above table. When we've finished building it, tell the copper where the list is, and the copper will execute this list every frame until we tell it to do otherwise.

I have created some macros to build copper lists, and I will explain their use briefly here.

```
struct CopperList Cl;
COPSTRT(&Cl); moves the 'cursor' to the start of the copperlist
COPMOVE(&Cl, dest, source); ...
COPHIT(&Cl, x, y); wait for beam pos...
COPEND(&Cl); ends the copper list...
```

If you use these macros instead of the system ones, you will create a copper list which we can feed the copper directly, and in much less time. (And we don't have to do any sorting or merging...)

To install the new copper list, just use the macro (my macro)

```
SETCOPPERLIST(&Cl);
```

the copper is now executing your list, and COMPLETELY ignoring intuition's screens and windows. (they are still active as they would be if you hadn't taken over the copper, but there is no way for them to be seen unless your copper list is specifically designed to do so.)

To return control of the copper to the system, we must find out where the system copper list lives, and use the same commands as before, fortunately for us, the address of the system copper list can be found in part of the IntuitionBase structure, (you must have intuition open before you can use this macro)

SETSYSTEMCOPPER; (all these macros can be found in cop.h)

this macro finds the system copper list and tells the copper to execute it.

A COMMENT ON DESIGNING COPPER LISTS

When you are designing your own copper lists for your own purposes, and you are using the method described in this article, you must remember that we are completely bypassing the system list. Don't set up some intuition sprite structures and use calls to GetSprite() and MoveSprite(), because the copper is not doing anything with the system list. You may open as many intuition screens as you want, but they will not appear until you write the necessary copper list to do it. The screens still exist in memory, but they are simply not being displayed.

The thing about all of this is that you have to specify almost everything YOURSELF. You have to specify the resolution, the width, the size of the display, and lots of other details that you never had to worry about when using intuition (I will go into these details later).

Note: when the copper is taken over, most registers are not reset, so any registers (like the color palette) will still be what they were when the system was in control, until you change them. One set of registers which are reset every frame are the bitplane pointers - these must be included in every copper list if you wish any bitplanes to be visible.

A COUPPLA WORDS ON HOW TO DO EVEN LESS WORK

After you have written a few copper demos, you will notice that most of the copper list stays the same from frame to frame, so there is no point in rewriting the entire list, now is there??? This is the reason for the set of PLATE macros I have devised:

```
struct CopperPlate p;
struct CopperList cl;
INITPLATE(&p); initializes the structure.
ADDPLATE(&p, &cl); adds a plate to the structure
USEPLATE(&p, &cl, x); moves the copper list 'cursor' to the position
marked with the plate.
```

For example:

```
INITPLATE(&p);
COPSTRT(&cl);
COPMOVE(&cl, COLOR00, 0x123);
/* the rest of the palette */

ADDPLATE(&p, &cl); /* this is plate 0 */
COPMOVE(&cl, BPLCON1, 0x1ff); /* this needs to be changed */
COPMOVE(&cl, BPLCON2, 32); /* so does this */
/* more stuff in here which stays the same */

ADDPLATE(&p, &cl); /* this is plate 1 */

```

COPMOVE(&cl,... something else which needs to be changed.

```
COPEND(&cl);
```

The above routine builds all of the copper list, and 'marks' two positions which need to be updated every frame or so.

Then in the actual loop of the demo, all you have to do is this:

```
USEPLATE(&p, &cl, 0); /* moves the cursor to plate 0 */
COPMOVE(&cl, BPLCON1, pos); /* changed it */
COPMOVE(&cl, BPLCON2, 76); /* changed this one too */
USEPLATE(&p, &cl, 1); /* move the cursor again, to plate 1 */
COPMOVE(&cl, ...);
```

```
SETCOPPERLIST(&cl);
```

And so we have completely updated the entire copper list and only needed to rewrite 3 copper instructions instead of the x amount in the whole entire list. (think of all that extra time you have saved again! (using the intuition copper lists doesn't even think about giving you this option))

A NOTE ON WHY THE COPPER LISTS ARE DOUBLE BUFFERED

The copper list we write is the exact same one that the copper is executing. The display would become a slight mess if we were making changes to the copper list and the copper hadn't finished reading the list, so I double buffer the copper list. What this means is that I have two copper lists, one which the copper is executing, and one which the program is updating. Each frame the roles are swapped.

To get a much better idea of how to achieve double buffering, examine the sample program.

Now that you have a fair idea on how to use the copper directly, I will describe what and how to actually achieve effects using the copper. (if you want a precise idea instead of a fair idea, have a look at the commented source, [but make sure you read this first, as it explains a fair bit which is assumed in the comments])

SOME TEXT AND DIAGRAMS ON HOW TO SCROLL HORIZONTALLY

The most important register that you need to know about for scrolling is BPLCON1, or the DELAY register, or the SCROLL register. Here is it's bit allocation (I have also included BPLCON0, and BPLCON2):

Bit #	BPLCON0	BPLCON1	BPLCON2
15	HIREG	X	X
14	BPU2	X	X
13	BPU1	X	X
12	BPU0	X	X
11	HOMOD	X	X
10	DBLPF	X	X
89	COLOR	X	X
88	GAUD	X	X
87	X	PF2H3	X
86	X	PF2H2	PF2PRI
85	X	PF2H1	PF2P2
84	X	PF2H0	PF2P1
83	LPEN	PF1H3	PF2P0
82	LACE	PF1H2	PF1P2
81	ERSV	PF1H1	PF1P1
80	X	PF1H0	PF1P0

KEY

HIREG = 640 mode
BPU = Bit Plane Use code 000-110 (none through 6)
HOMOD = Hold and Modify
DBLPF = Double playfield (PF1 = odd PF2 = even bit planes)
COLOR = composite video color enable (no effect on RGB)
GAUD = Genlock audio enable
LITE = Light pen enable
LACE = Interlace enable
ERSV = external resync
PF2PRI = playfield 2 (even planes) has priority over (appears in front of)
PF2P2 = playfield 2 (odd planes)
PF2P1 = playfield 2 priority code (with respect to sprites)
PF1P2 = playfield 1
PF2H = playfield 2 horizontal scroll code
PF1H = playfield 1

Wow! The only part of that we are interested in is PF2H and PF1H, the horizontal scroll code (also known as the scroll register, or the delay value, for the rest of this article I will refer to it as the delay). I will attempt to explain how these work -

imagine a line 1 pixel wide running down your screen.

like that. The delay is set to 0 on all of the display. Now, somebody changes the delay:

Delay = 0

Delay = 2

Delay = 1

See how it moves the information to the right when you increase the delay. Notice also that there are 2 delay registers in the above diagram - there is one for each playfield (when in dual playfield

mode) but when in normal mode there is one for the odd planes and one for the even planes. Because we want all of our planes to be aligned, we set both of the registers to the same thing. In lo-res mode, Delay can be set to anything from 0 to 15, and in hi-res mode from 0 to 7. (don't ask why, it just is.)

The more observant of the readers will have noticed that this won't let you scroll very far; it will only let you scroll 1 word, but alas this is plenty! When you have scrolled 1 word you can fiddle the bit-plane pointers. (if that last phrase frightened you, don't worry all will be explained).

12 of the system registers are bit plane pointers, and they point to where the video beam fetches what to display. You will recall that each register is 1 word in size, and a bitplane pointer needs 2 words, so this is why there are 12 registers for bitplane pointers (2 for each plane).

They are called BPLxPTH, and BPLxPTL, the PTH (PoinTer High) one being the most significant word, or the left one, and the PTL (PoinTer Low) being the least significant, or the right one.

An interesting thing to note about bitplane pointers is that they are always even, the rightmost bit is always ignored, so if you are fiddling with the bitplane pointers, you should always fiddle in 1 word increments.

If you change the bitplane pointers to move the bitplane in 1 word increments, you can combine this with the delay registers to get smooth scrolling.

When you increase the bitplane pointers by 2 (1 word), you are essentially moving the displayed bitplane left 1 word. If you set the delay to 15, slowly decrease it to 0, the bitplane will scroll smoothly (right to left) for a distance of 1 word, and to cover greater distances you just set the delay back to 0 when you increase the bitplane pointers by 2.

WHAT HAPPENS WHEN YOU REACH THE END OF THE BITMAP?

You must remember that memory is in a long line, like a piece of string, and a bitmap is like a wooden board that string has been wound around; each time the string circles around, it goes a little lower down, so that you get a surface of string. To

simulate moving the pointer to the bitmap (where to start winding the string) draw a picture on the string, unwind it, and then rewind it starting at a different position, the picture will still be intact, just moved around a little, depending on where you started. The important thing to notice is that when you reach the edge of the board and the string circles round to the front, it is 1 line lower down than the previous time.

What this means is that each time you scroll the entire width of the bitmap, you will slide down 1 line (if you are scrolling from right to left, otherwise you will slide up 1 line), and you must take this into account if you wish whatever you are scrolling to stay on the same line of the display. (see the sample program).

IF MEMORY IS JUST A PIECE OF STRING, HOW DOES THE COPPER KNOW HOW TO LINE IT ALL UP PROPERLY?

There are 2 registers specifically for this purpose, they are known as the modulo, and tell the copper how long the string on the back of the board is. (The length of the string hidden does not have to be the same length of the string visible [this is the one problem with the string analogy])

Again, there is 1 modulo register for each playfield (or the even and odd planes in 1 playfield mode), and they express the distance of the invisible string in bytes.

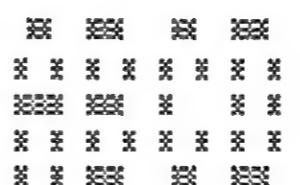
If you want the definition of the modulo in "technical" terms instead of my analogy, a modulo is a number that is automatically added to the address at the end of each line, so that the address then points to the start of the next line. Since they have separate modulus, the odd and even bit planes may have sizes that are different from each other, as well as different from the display window size.

For example, if you had a bitmap that was 40 bytes wide (320 pixels), and you were displaying the entire width of the bitmap, then the modulo would be set to 0, as there is no "hidden" string (memory). However, if you had a bitmap that was 80 bytes wide (640 pixels), and you were only displaying 320 pixels (40 bytes, probably in lo-res mode [but that doesn't matter]) then the modulo would need to be 40, to keep everything aligned.

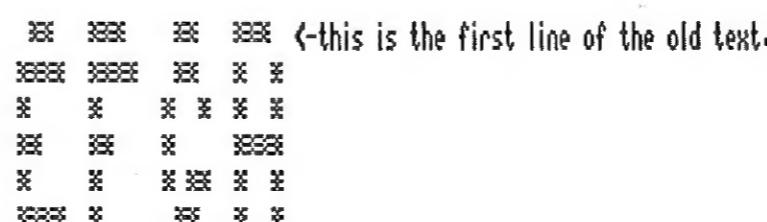
SEVERAL SENTENCES ON WHAT EFFECT THE ABOVE SECTIONS HAVE ON HORIZONTAL SCROLLING

When the text reaches the edge of the bitmap, and returns on the left-hand side 1 line further down, the top line of the text will have moved down one line in the bitmap (not the display). This means that the top line of the last bitmap worth of text will be 1 line above the "current" text. Essentially this can be viewed as garbage above the text.

A diagram of the first bitmap worth of text:



when it wraps around, and we write on some more text, here is what happens (we are writing in JAM2 mode, so the text is not layered).



notice the top line of the old text is above the new text. There are two ways to get around this problem:

1. Make sure the top line of any text you draw is blank (or uniform), which is difficult to do when using system fonts.

or

2. Make sure that the copper starts displaying at the start of the text, and not any sooner (this is the fastest way to do it, and is the method used in the demonstration program).

The second side effect of the "sliding" is that the area of bitmap being displayed slowly moves down through the bitmap (at the rate of 1 line per wrap around, to be precise). This means that we cannot scroll indefinitely using one bitmap; even the largest bitmap will eventually run out of lines.

I have found that the simplest solution for this is to put 1 screen worth of blank space at the end of the text, and when the scrolling area is completely blank, then clear the bitmap, reset all pointers, and move back to the top of the bitmap. Provided the bitmap has enough lines to cover all of the wrap-arounds in 1 playing of the message, it can be scrolled over and over with minimal fuss.

AN EXPLANATORY NOTE ON THE DATA FETCH AND ITS AFFECT ON THE MODULO

There are 2 registers which control the data fetch; DDFSTART and DDFSTOP. These registers control the horizontal timing of the beginning and end of the bitplane DMA display data fetch. The bitplane modulus are dependant on the bitplane horizontal size and on this data-fetch window size.

Standard DDFSTART and DDFSTOP:

Lo-res	\$38	\$40
Hi-res	\$3c	\$44

This is fine, but when you set the delay to something other than 0, you must fetch an extra word, so change DDFSTART to \$30 for lo-res, and \$34 for hi-res. (Also note that in the section of the display where the extra word is being fetched, sprites 6 and 7 are disabled (sprites 0 to 5 are completely unaffected). You must also reflect this change in the modulo - since we are effectively displaying an "extra" word, we just need to decrease the modulo by 2 (1 word)).

Recall before when we had an 80 byte bitmap and were only displaying 40 bytes, and we set the modulo to 40, if we wish to scroll this display, we must modify the data fetch to fetch an extra word, so we are really displaying 42 bytes, and therefore the modulo should be set to 38.

A FEW HINTS ON WHAT TO DO WITH THE SAMPLE PROGRAM

First, type it in (or download it, AmigaLink and Telecom permitting), compile it (I use Aztec 3.6a...), and get it working. Then, and only then should you start to fiddle with it. Try adding to the copper list: put in a few changes of background color, see if you can get the scrolltext to move up and down in a sinewave or something (hint: modify the COPWAIT before the COPMOVE to BPL1PTH), try to change the speed/direction of the scroll, multiple scrolls perhaps, ...in short fiddle all you want. Now that you understand the basic principles of the copper, have another look at the Amiga game loaders, and attempt to reduce them to their copper lists...

A BRIEF NOTE ON SYMBOL TABLE FILES FOR AZTEC USERS

If you have a look at the makefile, you will notice the argument +isym. This argument tells the

compiler to load a precompiled symbol table from the file name after the +i (in this case sym). The precompiled symbol table contains just what it's name implies, precompiled symbols. To create a symbol file, make a small C file with includes to every include file you can think of, and then compile with a +h(filename) option, the compiler will then compile all of the include files, and write out the symbols into the file you specified with (filename). I use a symbol file of all include files, and it is only about 100K in size, the size of the non-compiled include files is much larger, and it takes a lot less time to read in a 100K file than to compile the include files over and over (especially if the symbol file is in VD0:, along with the compiler, assembler, linker, editor, and linker libraries, (which is how I am set up...).

A BRIEF NOTE ON SYMBOL TABLE FILES FOR LATTICE USERS

I don't know if the new look lattice has pre-compiled symbol tables or not, I do know that the old look lattice didn't... Anyhow, if you don't have the option to use precompiled symbol tables, you will have to figure out for yourself which include files to use, 'cos I don't know which ones you need...

NEXT MONTH (MAYBE...)

Next month, if you're lucky, I might give you an in depth explanation (and some program) to add to this month's to move some sprites around. Using the copper directly to display sprites enables you to move the sprites EVERY frame with absolutely no difficulties what so ever, which means you get some VERY smooth sprites (ala DOC demo).

by Simon Bullen,
a Year 11 student
at Melbourne High School

Ed's notes:

Unfortunately, the program which accompanies this piece of prose was too long to print in one newsletter. So, if you wish to see the program he refers to throughout the article, you can find the program in the Australian public domain contributions area on Amiga Link I and II available for downloading. I apologise for the inconvenience. Also, Simon will be at the next main meeting, and he will be prepared to answer any questions on the above article, so look out for him...

Expression Systems	
Technical Writing & Computer Documentation	
Abssoft AC/Basic	\$250.00
Abssoft AC/Fortran	\$300.00
TDI Modula-2 Dev.	\$250.00
Metacomco Toolkit	\$100.00
MCC Pascal	\$150.00
Zuma Fonts Vol.2	\$50.00
Zuma Fonts Vol. 3	\$50.00
TV Text	\$150.00
SuperTex	\$50.00
Organise!	\$100.00
On-line!	\$100.00
Scribble!	\$100.00
Publisher Plus	\$200.00
Printer cable & driver for Citho	
Prowriter 24LQ	\$30.00
Sidecar Owners, upgrade with an IBM I/O card with 384k ram, clock, printer and serial ports	\$350.00
phone 232-3898	

Public Domain Update by Peter Jetson

Here's another load! Too bad some of this stuff didn't turn up over the Christmas break, isn't it....

Anyway, here's the list, order forms are on the inside back page or you can get them at the next meeting.

Amigan Disk #17

FontRepair Fixes a serious bug in CALLIGRAPHER font files created without another font as a template. Any Amiga will load such font files time after time until out of memory. See article in Vol. III, No. 3 of The Amigan.

Blanker2 A VERY small screen blanker. If you have always liked the idea of a screen blunker, but need all the memory you can get, this is the program for you.

DiskX Version 2.1 of this Disk Sector Editor is about the best of this genre I have seen for the Amiga. With this release, DiskX automatically scans your device list to see what drives you have on line.

DMEmacros A collection of Macros for DME (see Amigan Disk # ??) to

aid in program entry in your favorite computer language. (If you like C, Fortran, Modula-2, or Pascal) Plus some extra macros for general use.

DMouse Yet another Super Utility, this one lacks the clock, but does virtually everything else, in a LOT less space than most of the others out there.

ParalleLResource An example program to show how to use the Amiga parallel port.

PrintSpooler A TRUE background print spooler, that looks a LOT like those found on the VAX, etc. Even down to the style of the error messages.

PrivilegeHandler For those of us with 68010 CPU's, this file provides a means of booting SOME protected programs that don't like the 68010, by making a DeciGel-like program that will survive re-boot.

QMan A "Quick Mandelbrot" program. Good code examples including a couple of routines in Assembler, which is what makes this one "Quick".

Rokicki This subdirectory contains two others:

DFC A diskcopy/format program. Nice!

Life An interesting diversion, uses the blitter to make it BLINDLY fast!

TinyProlog A Prolog interpreter written in C. Nothing REAL fancy, but if you've ever wanted to try an AI (Artificial Intelligence) language, you can't beat the price on this one.

Zoo Version 2.00 of this alternative to ARC. by Rahul Dhesi, Amiga port by J. Brian Waters

ZorkLook A program to allow you to examine the text of all those Infocom games. (Ever wonder how they stored all that text on those small files? Try this and find out!)

Amigan Disk #18

DiskSalv Version 1.3 of the very useful disk recovery utility. There have been many enhancements over previous versions including hard disk compatibility.

Emit A program to transfer files across Amigas connected via null modem cable at very high speeds (up to 24k/sec).

PeliHacks Three display hacks by Oren Peli, author of Photon Paint. Source code is included.

RexXserial A library which provides serial device functions for ARexx. You must have the ARP library to use, and the program is in beta test version.

ScreenX - Put it in your Startup-Sequence and will provide you with a small clock/memory display, screen depth arranger, save a screen as IFF, or print screens. It also runs in the background automatically.

SGDemo This is the Strugen demo program. Strugen is a tool to generate appropriate optimized C source code of the contents of a window and its contents created with WYSIWYG efficiency.

Showfont Showfont will display all 256 characters of a font in a window. Handier than using Notepad or a font editor to see the contents of a font.

SimpleTerm- Source code and executable for a very simple terminal

program. The program is a usable and easily alterable assembler example.

TCB TCB will print a list of tasks and processes and other useful information. It will also display the code and data segments for a given process as well as many other structure-walking functions. Quite useful for debugging and learning.

QMouse A shareware program which provides a mouse accelerator, screen blunker, pointer blunker, automatic window activator (Sun mouse), hot keys, keyboard record/playback, and more with a cost of 4K!

QView QView is a shareware program also by Lyman R. Epp which opens a window and displays a file. It provides a search function and will also scroll left and right.

VirusX Steve Tibbet's Virus Removal System with coverage of the latest known viruses. VirusX will remove the virus it finds and stay in the background and look for more.

Amigan Disk #19

2000_Clock A text file detailing discussion about the real time clock in the Amiga 2000. Collected from Usenet.

Atom Clock A neat little program that will set the clock in your Amiga according to the Atomic clock at the U.S. Naval Observatory in Washington D.C. It uses your Hayes compatible modem and runs without much attention. It does forget to hang up the modem sometimes.

DME_v1.30c The latest version of a versatile programmers editor by Matthew Dillon. Adds environment variable compatibility, new invocation flags, ARexx port access, and more. Source and macros are included.

ReSource A demo version of an interactive disassembler complete with menus for the Amiga (with the SAVE functions disabled). This is the one for the true developer or hacker or aspiring programmer!

Amigan Disk #20

AZComm Version 1.00, a modification of COMM 1.34 added to implement ZModem protocol, 3/7 wire handshaking, and to work at rates over 19.2K bits/sec.

Eliza A version of the famous Artificial Intelligence simulation written in 1965 by Joseph Weizenbaum of MIT. Written in 68000 assembler.

FileInjector This is a super file requester which runs in the background and will 'inject' a filepath and name into the input stream of any program (which has a window)! It will sort by date, size, or alphanumerics. It is configurable and fun to play with.

HandYicons Extends the WorkBench with a new menu item which permits the user to run various Tools from the menu! Currently only Tools are supported.

HED 1.0 HED is an editor similar to ED, but enhanced with mouse functions, file requesters, and other useful items. There is not much other documentation except for the on-screen help

and menus it provides.

HodgEpodge An interesting exercise done by Robert Smith after an article by A. K. Dewdney in the August 1988 SCIENTIFIC AMERICAN. It will change a field of random pixels into patterns on the screen.

JimUtils A collection of Jim Cooper's favorite SMALL program utilities. Includes Avail, DFree, Freemem, Prefs, Scoot, Sweep, and Winsize. All were done in assembler and all are resident compatible as well.

LightninGlogger By Robert T. Shaw, a demo version of his file 'catalogger' which works for hard disks as well as floppy disks. This one keeps track of the filenotes and volume size and date information as well as the usual file info.

MultiView 2.0 A picture file viewer that will allow you to show Macintosh and Atari ST files in addition to the standard Amiga file formats. It will also allow you to save them in any of the above formats!

NewZAP 3.18 The latest version of John Hodgson's multi-purpose file/sector editor utility. You can edit a file in hexadecimal or ASCII sector by sector. Handy for those little patches and tweaks one needs to do occasionally.

Pointeranimator Includes two programs - the animator and the animation creator. The animator comes with some sample data to get started and a means to create your own using your own graphic editor.

Semaphore Tutorial A programming example in the use of exec semaphores. Brought to us by Rico Mariani from Usenet.

WesUtils Includes a word counter and a file to unlock the current directory and sets the lock to the root of the startup disk.

Templates The templates for all V1.3 commands, from Jim Butterfield. Asterisks mark all new and revised commands.

Syscheck A CLI program in assembler to check all revised Workbench disks for the final programs issued with V1.3.

Amigan Disk #21

AmNix111 Am_Nix V1.11 is an automated modem dialing system for Telenet's PC-Pursuit. The program will keep dialing the cities and destinations until it succeeds, or you get tired of waiting and cancel. Also included are the latest release of "The Final List," a shareware listing of over 1300 BBS's around the country AND overseas that support the Amiga, and a programs to convert "The Final List" file to a phone directory file for AmNix.

ASynch Sample C code by Phil Lindsay of Commodore-Amiga that shows how to do asynchronous file I/O with AmigADOS. Compiled with Lattice 5.0.

KeyClick A variable keyclick program for those who want an audible feedback from the keyboard. It features a slide gadget to control volume, and an ON/OFF switch.

Lucas The famous Public Domain 68020/68881 board for the Amiga! The author not only published this in The Transactor for Amiga magazine, he also uploaded it to several bulletin boards. Included is a set of modifications by Evan Sidorak to allow both the 68000 and the 68020 to live in Amiga at the same time. The directory is huge, for it contains not only

BBS All of the files necessary to create a bulletin board system called TAG-BBS. Documentation included.

ComMliNKBBS All of the files necessary to create another bulletin board system, this one called ComMliNKBBS. Documentation included.

Grep A utility for searching for strings and patterns in text files. C source included.

MenuEd1.2 A very handy programmers' utility for creating menus for programs. MenuEd allows you to create menus, see how they operate, and then write out C source code for inclusion in your programs. This version works properly under Kickstart 1.2. Documentation included.

Tek4010 A terminal emulation program which mimics a Tektronics 4010 graphics terminal, as well as a VT100 text terminal. This program also includes the Kermit and XModem file transfer protocols for exchanging files with bulletin board systems. C source included.

Devdisk #8

DiskPerf A suite of tests of disk performance which also can be run on non-Amiga machines for comparison. C source included.

FonEditor An editor for Amiga fonts. Documentation included.

HackGuide Text describing the tricks of the game Hack.

IffUtil Programs which read and decode IFF files. C source included.

MandFXP-D2 Shareware, very fast Mandelbrot set graphics display generator. The Mandelbrot is an infinitely detailed mathematical

object that lives in the complex number plane. This program can produce some wonderfully detailed images. Send away for the full version, which allows saving images to disk and printing of images.

MutuaLexclude Demonstration of the use of mutually exclusive gadgets. C source.

PowerwiNdemo Demonstration of the commercial program Powerwindows, which lets you interactively create windows, gadgets, and menus, and generates C code. This version is crippled-- it does not create any source code. Included because this looks like a very useful program for developers.

Pr Utility for spooling file printing so that printing occurs in the background. C source included.

QMouse Assembly source and executable. Used in Startup- Sequence file; queries the left mouse button and returns its status. Executable is only 104 bytes!

RamSpeed A benchmark program for evaluating memory speed. C source included.

Reassign A utility for reassigning sys: disk to another floppy or hard drive. C source included.

Shell2.04M Version 2.04M of Matt Dillon's command shell, which adds aliases, a history of commands, and other features to the CLI. This version includes command line editing using the cursor keys.

Text Assorted text files and notes.

Devdisk #11

AsmExample An example program written in assembler by Carolyn Scheppner of Commodore-Amiga. Demonstrates how to do some basic things in assembler, including interfacing to the library commands. Since this doesn't really do anything useful, I haven't included the executable-- this is primarily intended as an educational resource.

AsmRequester A file requester written in assembler. The assembled module is also included.

DirUtil5 Version 5 of the Directory Utility program. This program provides an alternate interface for manipulating files. C source included.

DOSHelper DOSHelper displays quick help summaries for each of the AmigADOS commands. C source and documentation included.

MandVect Graphics program which displays the iterations a point undergoes during the Mandelbrot transformation. Graphically interesting in its own right. Documentation included.

Polys Two simple polygon-drawing graphics thingies, one of which operates in hold-and-modify mode. C source included.

Touch Simple program to modify the date of a file. This version uses some AmigADOS magic (older versions just read a byte from the file and wrote it back). C source included.

Tracer Program which generates ray-traced pictures of a large number of spheres on a patterned plane. Ray-tracing is a graphic rendering technique which traces the path of each ray of light passing through or bouncing off of the objects in the scene,

producing a very realistic image. The spheres can be solid, transparent, refracting, etc. A sample picture, checker.iff, is included. C source and documentation included.

Devdisk #25

BlitLab1.2 Blitlab is a program which lets you experiment with the blitter to your hearts content, in relative safety. It opens up a Workbench window with gadgets for all of the registers of the blitter, and allows you to manipulate individual registers and perform blits on a magnified bitmap. This improved version handles line-drawing correctly. Documentation and C source included.

ClicKtOfront This utility modifies your Workbench environment so that double-clicking anywhere on a window will bring that window to the front.

Cmd This utility allows you to save the output going to the printer into a disk file. The file may be printed later with a command like "copy Savedfile to par.". Cmd works from both the CLI and Workbench, and can save the output from any program. Documentation, C & asm source.

Du-VI New version of the Directory Utility file manipulation program. This version opens two windows with lists of files and a third with all of the gadgets, so you can re-arrange your screen as you like it. C source.

Eless A very fast directory listing program. C source included.

IffReader A program for reading and displaying IFF graphics files. C source.

MonProc2 Monitors the activity of Amiga processes. This version has extra features for monitoring AmigADOS activity. C source included.

Munch A graphics demo called "Munching Squares". C source included.

Pipehandler New version of an AmigADOS handler implementing a pipe device. The pipe device allows you to take the output from one AmigADOS command and send it to the input of another command, without saving the output in a file first. Documentation and C source included.

PlacEwindow This program allows you to re-size and re-position a window. C source.

Psychic Cute fortune-telling program. C source included.

Query Two programs intended to be run from your Startup- Sequence. Query gets a yes/no response from the user. IfNorm checks for non-auto-configured ram installed on your system. C source included.

Rocket A lunar-lander type toy using sprites on top of your Workbench screen.

SmalLlib A small library serving as a replacement for amiga.lib for assembly language programmers.

StdFilEreq C source and test program for a standard file requester which you may include in your own programs.

Text Assorted text files and notes, including a description of the new IFF form for animated objects.

UnShar A program for extracting the files from a shell archive. Most software posted on the Usenet network is in shell

archive format. C source.

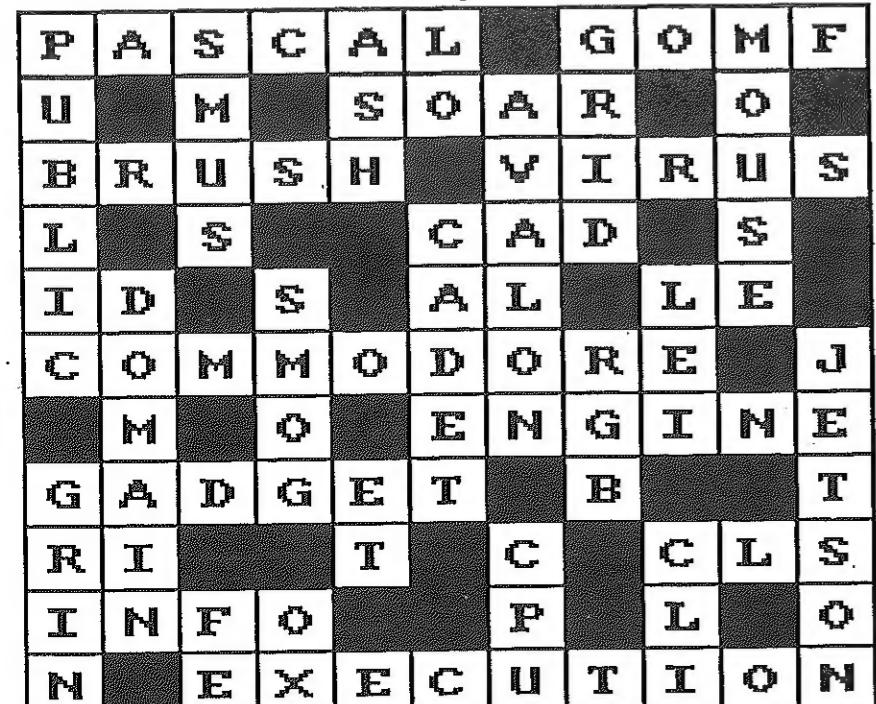
UseRcliPrects A modification of the Dotty demo showing the use of UseRcliPrects in C.

Viacom Another graphics wonder from Leo Schwab. This one demonstrates Leo's opinion of his favorite cable TV company. C source included.

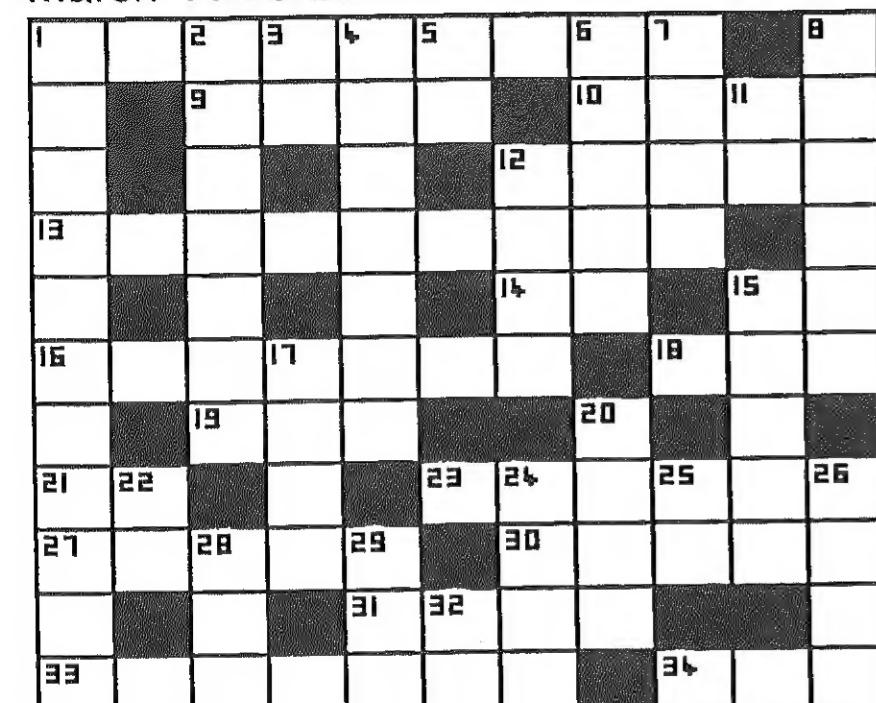
WarpText Mega-fast text drawing routines written in assembler by DevWare's Bill Kelly. Assembly language source and documentation included.

Amiga Word
by Alan Garner

Answers to February AWord:



March AWord:



Ok all you serious XWord and Amiga Fans, here is the next selection courtesy of Alan Garner. Alan has become a much appreciated regular for Workbench, and this brings up an interesting topic: If YOU have any great ideas for Workbench, (don't have to be great, a crossword has been done before) then send me a letter, give me a call or leave me a message on the boards. - Ed.

=====AJG 1989=====

ACROSS

1 The Scales and the Ram in conjunction for storing books and routines
 9 Rare one removed from 1A produces system directory
 10 A sip is up-ended for leaning tower location
 12 Transfers wines
 13 We hear this hi-res mode has kinky dressing habits
 14 Scale for determining acidity
 15 Royal pronoun
 16 William Eric and John surrounded Amiga's birthplace
 18 To sin in BASIC will return the code
 19 Record I/O in Brazil
 21 Special Stack Pointer
 23 Impish gel
 27 Zero talc was upset by the number system
 30 Lease went head over heels when used by artists
 31 Software for controlling the Amiga's tasks and interrupt vectors
 33 Video device that probably requires a skeleton key
 34 We hear the man is a colourful character (probably a bit shady)

DOWN

1 Supertramp tune for computers ? (2 words)
 2 Second class garbage makes the data manipulator
 3 Tokyo mile
 4 The endless slaughter of the unborn attempts to cancel an I/O request
 5 Initially the Royal Society is not too good
 6 Period
 7 Father
 8 Video display from right flower
 11 Saint
 12 Father repeating across the airwaves
 15 Western ceremony for file output
 17 Latvian capital
 20 Gray's language lost nothing for the BBS sysop
 22 The perfect computer for the police ?
 24 Have a quick look at this BASIC command
 25 The small island does exist
 26 Otherwise it makes provision for conditional program flow
 28 27A base plus two
 29 70's rock band confused the lion
 32 Could be an XT/PC hybrid ?

AmigaTeX

Quality desktop publishing at low cost.

Free demonstration diskette available; send a stamped self-addressed envelope.

All you have to do is the typing; TeX does many things which are part of high quality desktop publishing for you *automatically*.

LaTeX (which is included free) makes TeX as simple to use as any word processor for your general correspondence.

AmigaTeX is a complete implementation of the TeX typesetting system created by computer legend Professor Knuth of Stanford University. A WYSIWYG screen previewer is included. Output may be printed on laser or dot matrix printers (ring for details of printers supported).

AmigaTeX \$612 (inc. tax)
 Metafont \$198 (inc. tax)

TeXworks PTY. LTD

78 Nott Street, Port Melbourne, Victoria 3207
 Telephone: (03) 646 5613

The Author of the article on the opposite page is Alex Warman as you have probably gathered from Texworks inc. He has kindly offered to submit a regular (1 page) article in Workbench on Desktop Publishing, and many of the aspects and techniques involved in that area of home computer usage.

REASSIGNING SYS:

or
 Hacking for Beginners
 by
 Mark Kelly.

The disk you insert after the upside-down hand asks for the Workbench disk is much beloved by your Amiga, especially when you are using the CLI (Command Line Interface). Whenever you give a command from the CLI, your Amiga first turns to that disk to find out what it needs to know. The Amiga calls that disk "sys:" (the "system" disk) and expects to find everything on it unless you tell it otherwise. Even if you never use the

CLI, your Amiga still often looks at the system disk to find information and if the system disk isn't there, you could be in for a lot of disk swapping. There are two main types of things the Amiga will want to look up: commands and information.

1. COMMANDS

When you give a CLI command, the Amiga searches it's "path": a list of places where the command might be found. It will look in the root directory, then the C directory (where most AmigaOS commands are kept). Failing those, it may then try the SYSTEM directory and if it still hasn't found the command, your Amiga gives up and says it doesn't know that command. To find which places your system will search, get into the CLI and type PATH. If the command you want to run is not in one of those locations, there are three common ways of getting the Amiga to find it.

(a) Give a full pathname. If you know the command is on the disk in df1: you just tell it that. You type df1:command. If the command's in the C directory of your RAM: disk, you enter ram:command. If it's in the Z directory of the disk named "Fred" you type Fred:Z/command. The typing before command in each case is the path.

(b) Get into the proper place. If the command's in the FRED directory on df1: you can make that the current directory with CD DF1:FRED. Since the Amiga always searches the current directory first, it will find the command. This method is handy if you plan to frequently play with a few commands in a known directory. The drawback is that the path is now limited to that directory. Even if you enter "c:dir", the Amiga will complain because there's no C sub-directory in the directory you're in. You have to use sys:c:dir to specify the full path.

(c) Extend the path. The best way! The PATH command (introduced in Workbench version 1.2 - how did we survive without it in the original version?) redefines the search path. You can add any directories, devices or disks to the search path, e.g. PATH RAM: ADD tacks RAM: onto the list of directories to search. My s/startup-sequence is something like: PATH RESET vd0:c, vd0:, ram:, ram:c, system, 's (the operating system automatically adds sys:C and sys: to the path you give). I usually keep my most commonly-used CLI commands in vd0: (the superior recoverable-ram disk from ASDG: it's better than RAD: offered in the 1.3 upgrade because it shrinks and expands

The Publisher's Corner
Hyphenation and Justification

Alex Warman, *TeXworks Pty. Ltd.*

As the term *Desktop Publishing* is widely-used jargon, but only vaguely understood I have submitted this article which discusses some of the micro-details of typesetting as distinct from typewriting or word processing.

First a disclaimer: while our company sells AmigaTeX (in fact TeX systems for IBM PCs, Macintosh, and other machines) these article is not a promotion for TeX. However, sometimes I will explain the way TeX behaves because Professor Knuth has made a close study of the *old craftsmen* typesetters and tried to make TeX mimic their behaviour. For any-one who doubts this I recommend you refer to *The TeXbook*, Knuth (Addison-Wesley).

Well, what does Desktop Publishing mean? It is used to cover a multitude of sins. Publishing can mean anything from eye-catching leaflets with drawings, photographs or cartoons and some words here and there, to a fiction book, or a technical report with mathematical and scientific notation. Desktop simply means that your publishing can be done on a computer which is small enough to sit on a desk top; Amigas certainly qualify, but so do IBM PCs, Macintosh's, Sun and Apollo workstations...

In my opinion, *Desktop Publishing* does not mean anything specific; you need to look carefully at the characteristics of what you want to publish before deciding which software tool is well-suited to *that sort of publishing task*.

Why are **Hyphenation** and **Justification** important in typesetting? For those used to typewriters and more recently word processors, you will be used to seeing *non-proportional spacing*. This means that each character occupies the same amount of space. Each word is separated by at least one space character (the same width as each letter).

While a clean right margin is usually required, authors are not very cooperative in writing words so that each line finishes in the same place. To get a clean right margin extra space characters must be put between some words.

In commercial publishing the sets of characters used (called typefaces or fonts) have different widths for various characters; this is known as *proportional spacing*. Getting a clean right margin now, is a much more complicated job as the inter-word spacing is not made up from a single size of space. In fact in typesetting there are quite complicated ways to decide the

limits for inter-word spaces based on various things including the choice of typeface.

Some examples show different widths using the same typeface, Computer Modern (CM) typewriter, roman, italic and bold.

Encyclopedia - non-proportional typewriter font.

Encyclopedia - proportional CM roman font.

Encyclopedia - proportional CM italic font.

Encyclopedia - proportional CM bold font.

This fiddling with inter-word spacing is called **justification**. Why go to all this trouble to do the sort of manipulation described above? It turns out that when people read for extended periods, eg. when you are reading a fiction book, it is much easier to concentrate for longer or to read faster if text justification (and also hyphenation) has been done well, ie. the way the old typesetting craftsmen used to do it. As most long books do have a clean right margin these problems must be dealt with.

It turns out that trying to stretch or squeeze the inter-word spacing to get a clean right margin is not always adequate. Then you must decide if a word near the end of a line should be hyphenated to keep the inter-word spaces within the required limits. The procedures, used for deciding how to hyphenate a word, vary enormously in sophistication. For anyone interested TeX uses a method developed by Frank Liang whose Ph. D. work studied all known methods in order to find the best procedure; a brief description is in *The TeXbook*, Appendix H.

Hyphenation is a complicated problem, but without it some lines would have very large space between words which can look ugly. If it is done badly you might get effects like "the-rapists" or "pre-ached"!

Justification and hyphenation become even more problematic as the width of the page narrows; even worse in multi-column formats like this magazine. As column width narrows there are fewer inter-word gaps where the space can be fiddled with. Thus any system will hyphenate more, but too many hyphenated lines look ugly too; so what to do? As with many things in life you just have to comprise.

One thing in TeX's favour as a typesetting system is that its justification and hyphenation behaviour can have their limits varied. For instance in my two-column article in the December issue the justification was much tighter (but more hyphenated lines!) than the limits used for this article. So, this article has few hyphenated lines for a narrow column format. For anyone interested in the gory details, see *The TeXbook*, Chapter 14, How TeX Breaks Paragraphs into Lines.

according to need. RAD: always chews up all the memory you allocate it whether you're using it or not. [but vdk is *the* best RRD!]

2. INFORMATION

Commands, therefore are pretty easy to find. It's not so with information files needed by the system. With these files (like "library" files and "device drivers" which have data to control devices like the printer, serial port or speech synthesizer), the Amiga is less talented at finding things. It expects them to be in certain places (e.g. the LIBS directory) and if they're not there, the system won't look anywhere else and it will give up. If, for example, you remove your Workbench disk and want to use the SAY TRANSLATE\$ command in AmigaBasic, the system will want to read files such as translator.library in the LIBS directory and narrator.device in the DEVS directory. That's why you'll get a requester asking you to insert the WorkBench disk. Even if you insert a disk with the required files on it, the Amiga won't accept it if it's not the same disk you booted with. Even an exact duplicate of that Workbench disk won't do [unless you use a backup-program that doesn't update the disk's creation date]. I told you the Amiga loved the boot disk! To get the Amiga to accept another disk (or device such as RAM: RAD: VD0: or DH0:) as the system disk, you have to tell it. This is what the following batch file accomplishes.

A practical example: you booted with WORKBENCH and want to swap to WORDPERFECT which is on its own bootable disk. ("Bootable" means it can be inserted after the KICKSTART disk when you're turning on or resetting the machine. It contains the necessary files for the system to start up). If you pop out WB and slip in WP, get into the CLI and type "dir" you'd expect a directory listing, right? After all, the WP disk does have the DIR command in its C directory, just like WB. Instead, you get a requester asking for WORKBENCH to be replaced. Why? As far as the Amiga is concerned, the place to find the command is still the WORKBENCH disk - the one you started with. Short of resetting, you have to have to eject the new disk and replace WORKBENCH to do just about anything.

This is where this batch file comes in. It makes the Amiga forget about the disk you booted with and form an attachment with a new disk (or device) as if you had booted with it in the first

place. (It's a batch file, which assumes you can at least get into the CLI, type in the batch file and save it. You can't? You're not a REAL Amiga user then. Skip to the next article in this illustrious publication, buy an AmigaDOS manual and start reading!)

Basically, the batch file called SYS reassigns a new disk (or device) as the sys: disk. Once it's EXECUTEd, as far as the Amiga's concerned, the new disk is the one you booted with. You can merrily leap from any system disk to another without wrist-wrenching requesters demanding disk changes. If you want to use a hard disk or memory "disk" as your system disk, this will arrange it.

Type it in, store it as "SYS" in the S directory. If you have upgraded to version 1.3, type PROTECT S/SYS +s so you can run it by typing S/SYS (or just SYS if your path includes S) rather than EXECUTE SYS.

```
.key device
; all typing after semi-colons are comments (like this)
```

```
; if device was not given, complain & exit
CD sys:
If <device>Q EQ "Q"
  Echo "Usage is SYS device: (e.g. SYS df0:)"
  Quit
EndIf
```

```
Copy c:assign ram: ;get a copy of ASSIGN to use
```

```
Ask "Insert new sys: disk (if necessary) & hit RETURN"
; The ASK command is only in version 1.3.
; It pauses to let you swap disks.
; If you don't have the 1.3 upgrade,
; use the following 2 lines instead,
; WITHOUT THE SEMI-COLONS!
```

```
; Echo "Insert new sys: disk (if necessary) & hit RETURN"
; RAM:Assign >nil: ?
```

```
; Whenever <device> appears, the Amiga will substitute
; the device name you gave in the command line.
```

```
ram:Assign c: <device>
ram:Assign system: <device>System
ram:Assign l: <device>l
ram:Assign devs: <device>devs
ram:Assign s: <device>s
ram:Assign libs: <device>libs
ram:Assign sys: <device>
; put other necessary ASSIGNments here
; e.g. fonts, env, t, clipboards
```

```
; The new sys: disk is now in control.
; It should have DELETE, CD and ECHO
; in the C directory for this to be able to finish.
; If it doesn't, copy them to ram too and use the
; ram: copies (e.g. ram:Delete... ram:CD... etc.)
Delete ram:Assign ;tidy up
CD sys:
```

```
Echo "Sys: reassigned to <device>"
```

```
; SYS batch file by Mark Kelly (13-Jan-89 00:27)
; reassigns sys: to another bootable disk (or device).
; Format: EXECUTE SYS <diskname or devicename>
; e.g. SYS df0: SYS WP: SYS RAD: or SYS VD0:
```

AMIGA DRIVES

**VDrive3.5 1/4 Height 3.5" NEC Mechanism
With Pass-Thru and Drive
Disable Switch.**

**VDrive5.25 1/2 Height 5.25" NEC Mechanism
With Pass-Thru, Drive Disable
Switch and Write Protect Switch**

**\$240 each...For a limited time only. \$240 each.
(Both covered by 10mth warranty.)**

**Contact Peter at: P.M.Developments on
018 322 920 or 749 4128**

Co-ordinators Comment

I have to begin this month's AUG co-ordinators report by announcing the resignation of the current co-ordinator, Bob Scarfe. Bob is leaving AUG and indeed Australia for the next 12 to 18 months. As Assistant co-ordinator I will be taking on his duties until our annual elections in June/July.

On behalf of the committee and members of AUG I would like to thank Bob for the time and effort he has put into representing and organizing our group over the past two years. Bob, we wish you well in your new ventures and hopefully you can send us an occasional report on Amiga activities as you travel...

A number of other changes have occurred recently within AUG and the committee has begun the process of examining what direction AUG should take as it nears it's third birthday. For those of you who have not been with the group from the early days AUG held its first formal meeting in March 1986. Since that time we have grown considerably

and the interests of our members have changed. We hope to be able to follow the changing needs of the members of AUG as that is the whole basis of a group such as ours - to provide support in whatever ways possible for owners and users of Amiga computers.

It is not possible, however, for a committee to meet the needs of its members without some form of feedback and assistance from the members in general. There has been constructive criticism recently from concerned members and the group will gradually respond to many of the (welcome) suggestions. Some of the decisions made at recent committee meetings we hope will benefit AUG members and the Amiga community.

- AmigaLink II is to be upgraded to 2400 baud. This will occur as soon as purchase of a suitable modem can be arranged.

- A 'flyer' is being produced for distribution to Amiga dealers etc. to advertise the existence of AUG. This had been delayed until we had established regular meetings at Burwood State college.

- AUG is to investigate publication of special topic booklets to cover areas of interest beyond what would normally be published in our monthly newsletter. Our editor is currently preparing a draft on Amiga viruses with contributions from a number of sources. This will be produced in a similar format to our newsletter but will be sold/distributed as a publication in its own right.

- In future, Amiga dealers will be invited to display and sell their products at AUG meetings. If you are interested in selling Amiga related software/hardware at AUG meetings please contact our purchasing officer (Drac) and he will organize this. If sufficient interest is shown a separate room can be set aside especially for traders.

- Copying of Public Domain disks is once again being done at the monthly AUG meeting. This had been discontinued because it required the software librarian for the whole of the meeting. Thanks go to Craig for volunteering to start this service up again. Be patient with your orders, it is not always possible for the copying to be done on the day.

AUG is also currently setting up a user HELP-NETWORK with the aim of providing contact between members requiring assistance and those

able to help. This service will be entirely voluntary and is not intended as a substitute for having and READING the proper manuals. If this service is abused it will be discontinued. More details regarding how this service will operate and how to get help will be published in next month's newsletter. If anyone is interested in volunteering their services please contact me. I thank those who have already registered their offers of assistance and I am pleased so far with the response - but don't let that stop you volunteering if you have any special area of knowledge/expertise.

Commodore have made good their offer of Professional Page and Con is currently evaluating its suitability for preparing our newsletter, I expect we will hear more from him on this subject. Commodore are to be congratulated for beginning to support the Amiga user groups and realizing the important part we play in the success of their products. I trust we will see more signs of interest from them in the future.

The February main AUG meeting had the usual 150+ member turnout that is typical of recent times. This is not a large attendance considering the size of our group but for those who did attend there were some worthwhile demos. Norm Christian supplied a disk of Amiga art which was quite impressive on the projection screen and made excellent use of colour cycling. Ben from Ultraphase (Knox) had a 'Flicker Fixer' set up on an Amiga 2000 with a high-res monitor, it was displaying interlace and hi-res pictures completely free from the usual annoying screen flicker. They also had a number of new software releases and a high quality micro-switch joystick.

There were several interesting questions raised in the general Question and Answer section, one which raised the possibility of setting up a 'classified adds' section in the newsletter for members who wish to sell/trade. The committee will investigate the requirements for this so look for further news. There is always a notice board available at the meetings and if time permits announcements can be made in the main meeting. A group from Latrobe University mentioned the existence of some 40 megabyte Amiga related P.D. software online and available for public access, some of which had not yet made it to Fish disks. Registration may be required, more details will be published later. A representative of a Commodore users group from Knox was at the meeting and is interested in setting up an Outer Eastern Amiga SIG - more details next month.

Some of the Special Interest Groups (SIGs) were low in numbers, it may be time to restructure some of their activities and perhaps combine them to form stronger groups with broader interests. One new SIG, the Hardware group is still seeking a formal co-ordinator but had a big following and seemed to have plenty to discuss. Justin Summers put together a presentation on adapting bare 3 1/2 and 5 1/4 floppy disk drives as external expansion drives on the Amiga. He explained the logic required and has made available a kit in the form of a PCB with documentation. See him at meetings for further details. Next month the group plan to present details on a hardware battery backed clock for the Amiga 1000. This group is likely to combine with what was originally the Developers SIG because of common interests.

At the March meeting we will have a demonstration of 'GENP' a Genealogy/family history package for the Amiga. It is to be presented by the developer, Peter Evans, and is also likely to be the subject of a future newsletter review. GENP is one of the now growing range of software from Australia. We have also had a request for Con to give a demonstration of Excellence! as he speaks so highly of it as well as using it to produce Workbench each month. Perhaps April? If there is anything you as members of this group would like to see happening at our monthly meetings please talk to one of the committee members about it.

Lester McClure,
Feb 89.

SCRAMBLES
(Assortments of Con's RAMBLES)
by Con Kolivas.

A lot of people have told me they like the new format of this column, so I will keep it that way... At least if there are things to talk about anyway.

So, what's new?

The scanner I mentioned in last month's scrambles had the wrong model number quoted. It was the JX-450 from Sharp along with the ASDG software. I have already had one person phone me asking if I had more information on it. Alas, I have said all I know. This person did say that the AT he updated to from his old Amiga would probably be updated to a 2000 (you know what I mean, he's

going back to the best) due to the software available for this scanner. If he's impressed he might be able to offer a demonstration at one of our meetings! I'm sure we all appreciate the offer, but I apologize in not catching his name...

We are about to be inundated by *heaps* of new software in areas new to the home-computer world for the Amiga - in particular lots of animation and ray tracing/modeller type programs (which will you choose?)

The UNIX™ based Amiga 2500 will be released soon with the 68020 and 68881 co-processor. This is not the news. The news is that Commodore have taken too long! (Well that's yet to be seen). The reason: the new read/write many compact disc type storage systems (NOT worm drives) are being released in the States for \$7,000 US. How that relates is the company releasing them is also releasing a UNIX based system with this storage system (gigabytes), running a 68030, 68882 and it's own OS on the side (the 2500 will be UNIX only). What's worse (or better whichever way you see it) is that it will sell for \$6,500 US. How they'll do that I don't know. It will be called (appropriately) "NEXT".

The 2000 now only sells at retailers as a complete package - 2 drives, monitor and \$3,000 worth of software. This will all go for \$5,000 AUS. The reason: people who have 5 grand or more to spend are not going to buy a 3,500 dollar computer, so they would go for the MAC. How this works is that having a 5 grand system that is much more appealing should outshine the MAC for the Layman buying a business computer. The logic may sound ridiculous, but Commodore are sticking with it. (Why they don't sell it both ways I'll never know??)

Commodore have recognized the 500 as their winning piece of hardware, so they are finally developing some expansion hardware for it. The A590 hard drive for the 500, yes the 500 is being released soon. It will feature 20 megs of disk storage space, AND room for plug in 1 meg chips! Great stuff...

Very hush hush (rumour only...). I've heard on the boards that DPAIN III is coming! Extra features: EHB, auto PAL detect, and an animation form! Who knows what else it might have (if it comes out of course)...

Editor's Column
(Written February 28, 1989)

Thank you to Mike Clark, representative from Commodore Sydney, for offering and sending the Amiga Users Group a copy of Professional Page. I will get in touch with you as soon as my phone bill drops to a respectable level (these holidays seem to overwork the bulletin boards and the phones...)

As for Professional Page - it looks great! The manual is very easy to read and introduces you so quickly to everything that you will be surprised how powerfully you can command the program with so little practice. So what was this month's newletter done with? Professional Page was used for the cover inside and out, but Excellence! was used for the text... NOT because I preferred Excellence! and am a creature of habit (on the contrary), but by the time I reached the fourth page of the newsletter, my poor old 1 meg machine ran out of memory. So, until I get some extra memory which is difficult for a full time student like myself to afford, then I will use it for it's ability to create the fairly complex covers, rather than sticking them together with art glue.

I forgot to mention last month (in my hurry to complete the newsletter on time) where the cover picture came from. It is called Pinball, and it is from the Deluxe PhotoLab art disk. This month, a beautifully digitized picture of a Lion from the demo images of Pixmate - this one goes into the Hall of fame for me!

I have had some people question the reward system I use for this newsletter. Let me say that I am fairly aquainted with human nature and know from the evidence that the majority of contributors like it. However, if you wish to contribute and not receive a reward other than that involved in seeing your article, I would be more than glad to print it without sending any tokens, and will give you a special mention if you don't mind.

Speaking of special mentions - Darren King, our writer of the infamous Virus Article, will be the main author for the anti-virus publication, and it will document details of some 12 viruses or strains, and good protection programs worth noting.

I quite like the new size of the newsletter and hope you do too. I'd like to thank all contributors and future contributors for their efforts and hope to see the same support. See you at the next meeting

Public Domain Software Order Form

Mail to: Amiga Users Group, PO Box 48, Boronia, 3155, Victoria

Disk Numbers:							
Don't forget to specify collection name, ie Fish, Amigan, Amicus, etc							
Disks supplied by Amiga Users Group @ \$8 each							
Disks supplied by member @ \$2 each							
Club Use Only:							
Member's Name: _____ Membership #: _____							
Address: _____							
Postcode: _____							

Newsletter Back Issue Order Form

Mail to: Amiga Users Group, PO Box 48, Boronia, 3155, Victoria

Issue Numbers:							
Be patient, we may have to reprint some issues to fill your request							
Number of issues ordered @ \$2 each							
Club Use Only:							
Member's Name: _____ Membership #: _____							
Address: _____							
Postcode: _____							

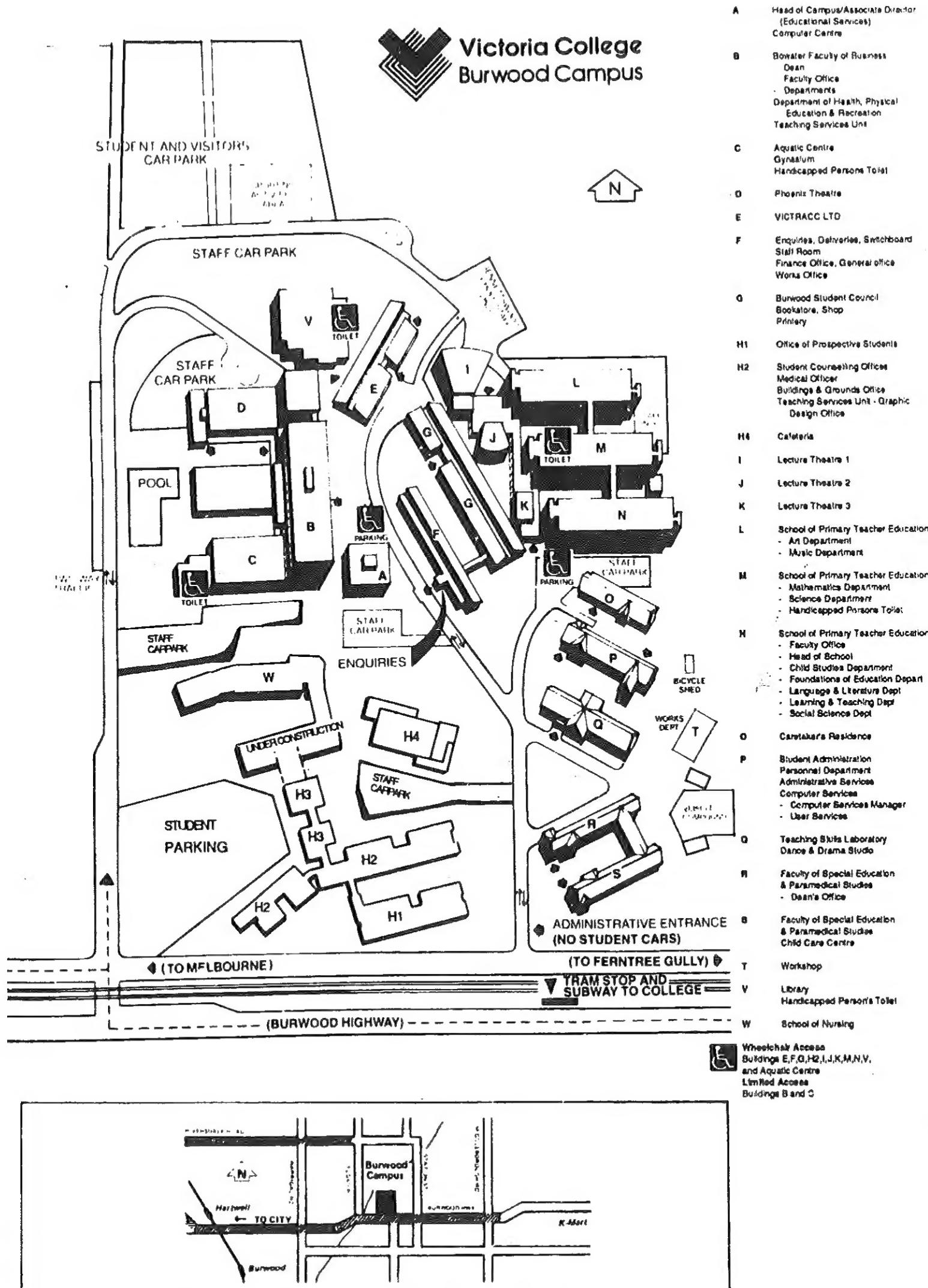
Application for Membership of The Amiga Users Group Inc

Membership is \$25 per year. Send your cheque to: Amiga Users Group Inc, PO Box 48, Boronia, 3155

Surname: _____		Details on this side are optional	
First Name: _____		Year of birth: _____ Which model Amiga: _____	
Address: _____		Occupation: _____	
Postcode: _____		Interests: _____	
Phone Number: _____ STD Code: _____		Dealer's Name: _____	
Where did you hear about AUG: _____		Dealer's Address: _____	
Signed: _____ Date: _____		If admitted as a member, I agree to abide by the rules of the Association for the time being in force.	
Club Use Only	Date	Paid	Receipt #
Memb #			Card Sent

March 1989 Amiga Workbench

**AUG meets on the
third Sunday of each month**



Where is Victoria College, Burwood Campus?

People often have difficulty locating our meeting place the first few times. Victoria College is on the North side of Burwood Highway, Burwood, just East of Elgar road. Coming from the City along Burwood Highway, turn left at the first set of traffic lights after Elgar road. Follow the road around past the football oval, over three or four traffic bumps to the car parking areas near the netball courts. Further up the road, to the left, you'll find Lecture Theatres 1 and 2.

If you have a Melways, try Map 61 reference B5.